PUC Minas

Linguagens de Programação (Ciência da Computação)

Professora: M. Sc. Luciana De Nardin
luciana@pucpcaldas.br

1

Histórico das linguagens de programação

- **1936 – 1945 – Computadores eletromecânicos**
  - 1945 - Alemão Konrad Zuze cria a linguagem de programação **Plankalkül** $\Rightarrow$ programa de cálculos
  - Nunca foi implementada
  - 1972 $\Rightarrow$ data de divulgação da linguagem
  - Estruturas de dados avançadas
  - Aritmética em ponto flutuante
  - Estrutura de repetição semelhante ao *for*
  - Estrutura condicional *if* mas, não tinha *else*

- **Exemplo:**

```
P1 max3 (V0[:8.0],V1[:8.0],V2[:8.0]) => R0[:8.0]
max(V0[:8.0],V1[:8.0]) => Z1[:8.0]
max(Z1[:8.0],V2[:8.0]) => R0[:8.0]
END

P2 max (V0[:8.0],V1[:8.0]) => R0[:8.0]
V0[:8.0] => Z1[:8.0]
(Z1[:8.0] < V1[:8.0]) -> V1[:8.0] => Z1[:8.0]
Z1[:8.0] => R0[:8.0]
END
```

- **Final década 40 - início década 50**
  - Computadores lentos, pouco confiáveis, caros e memórias extremamente pequenas
  - Não havia linguagem de programação
  - Não havia linguagem de montagem

```
IA-21 (10110000 01100001) ⇒ MOV AL, 61h
```

- **1949 – John Mauchly – *Short Code***
  - Rodava em: BINAC – UNIVAC I
    (1951 – 1º computador comercial)
  - Linguagem interpretada
  - 50 vezes mais lento do que o código de máquina

Ex: `X = SQRT (ABS(Y))`

Em Short Code:

```
00 X 03 20 06 Y
```

- **1954 - FORTRAN**
  - ***FOR****mula* ***TRAN****slator* (produto IBM)
  - Decorrência do desenvolvimento do IBM 704
  - Primeira linguagem compilada
  - 1958 – metade dos programas eram escritos em Fortran
  - Memória disponível ≅15 kb

- **1954 - FORTRAN**
  - Alocação estática de memória
  - Bastante utilizada até hoje…

Avaliação:

- Domínio: programas numéricos por natureza
- Mudou radicalmente e para sempre, o modo de utilização dos computadores

- **1958 - LISP**
  - John McCarthy (MIT)
  - Necessidade: processamento de dados em listas (não em *arrays*) e computação simbólica (em vez de aritmética)
  - Toda a computação até então era voltada para dados númericos em matrizes
  - Dois tipos de estruturada de dados: átomos e listas

Avaliação:

- Domínio: aplicações em IA

- **1958 - LISP**
  - Toda computação é reduzida à aplicação de funções a argumentos
  - Sem variáveis ou atribuições
  - Controle por recursividade e expressões condicionais
  - Sintaxe C → mistura de inglês e algebra
  - Sintaxe LISP → lista e parênteses

Avaliação:

  - 1ª linguagem funcional
  - Dialetos: Scheme (1975), Common Lisp (1984), Haskell (1990)

- **ALGOL 58**
  - Metas (União ACM e GAMM):
    - Proximidade com notação matemática
    - Boa para descrever algoritmos
    - Traduzível para código de máquina

- **ALGOL 58**
  - Características:
    - Formalização do conceito de tipo
    - Eliminou nomes de comprimento arbitrário
    - *Arrays* com qualquer número de dimensões
    - Parâmetros separados por modo (*In* e *Out*)
    - Separador de comandos ‘;’
    - Atribuição ‘:=’
    - *If* com *else-if*

- **ALGOL 60**
  - Novas características:
    - Estrutura de blocos (escopo local)
    - Passagem por valor e por referência
    - Recursividade
    - *Arrays* dinâmicos
    - Sem manipulação de *strings*

- **ALGOL 60**

Avaliação (sucesso):

- Foi a primeira vez que um grupo internacional tentou projetar uma linguagem de programação independente de máquina
- Foi a primeira que teve a sintaxe descrita formalmente (*Backus-Naur Form* → método mais usado para descrever sintaxe de LP's)

- **ALGOL 60**

  Avaliação (fracasso):

  - Nunca largamente utilizada
  - Muito flexível >> difícil implementar
  - Resistência do FORTRAN

- **COBOL**
  - Domínio: aplicações comerciais
  - Ambiente = AIMACO (Força Área EUA) + FLOW-MATIC (UNIVAC) + COMTRAN (IBM)
  - Grace Hooper
    - Convencer programadores que um computador poderia entender palavras inglesas
    - Protótipo de uma linguagem que compilava código escrito em inglês, francês e alemão

- **COBOL**
  - Características:
    - Permitiu que os nomes fossem significativos
    - Introduziu a possibilidade de geração de relatórios

- **COBOL**

  Avaliação:

  - Comandos de seleção aninhados
  - 1ª linguagem de programação imposta pelo U.S.DoD
  - Atende bem a seus objetivos
  - Contribuiu para o desenvolvimento de compiladores

- **BASIC**
  - 1963 – ***B****eginner's* ***A****ll-purpose* ***S****ymbolic* ***I****nstruction* ***C****ode*
  - Voltada para estudantes não voltados para a ciência
  - Metas
    - Fácil de aprender
    - Agradável e amigável
    - Tempo do usuário é mais importante do que tempo de computador

- **BASIC**

  Avaliação:

  - 1º método amplamente usado de acesso remoto a um computador
  - Dialetos do BASIC: QBASIC (1989), Visual Basic (1990), VB.Net (2001 .. 2015)

- **PASCAL**
  - 1963 – Niklaus Wirth
  - Objetivo: linguagem de ensino

Avaliação:

- Simplicidade + expressividade
- Turbo Pascal

- **C**
  - 1972 – Dennis Ritchie (*Bell Laboratories*)
  - Objetivo: linguagem de programação de sistemas portáveis
  - Unix foi escrito em linguagem de montagem

- **C**

Avaliação:

- Instrução de controles adequadas
- Facilidade de estruturação dos dados
- Conjunto rico de operadores
- Ausência de verificação de tipos >> insegurança
- Popularidade >> fez parte do Unix

- **PROLOG**
  - Programação lógica é o uso de notação lógica formal para comunicar processos computacionais a um computador
  - Notação >> cálculo de predicados
  - 1972 – primeiro compilador

Avaliação:

- Poucas instruções >> complexa
- Formada por fatos e regras

- **ADA**
  - 1982 – Homenagem a Ada Lovelace
  - ADA = COBOL + BASIC
  - Projeto mais caro do U.S.DoD
  - 1983 – *Ada Language Reference Manual* (ANSI)

- **ADA**

  Avaliação:

  - 1ª linguagem OO padronizada internacionalmente
  - Encapsulamento (abstração de dados)
  - Manipulação de exceção
  - Programas concorrentes
  - Projeto compartilhado
  - ADA 95 >> U.S.DoD não exigiu mais sua utilização

- **C++**
  - Evolução de C (**C** **C**om **C**lasses)
  - Checagem de tipos, classes, construtores, destrutores, sobrecarga de operadores
  - C++ = C + Simula 67/Smalltalk

- **C++**

  Avaliação:

  - Popularidade >> muitos usuários de C
  - Relativa complexidade

- **PYTHON**
  - 1991 – Guido van Rossum
  - Desenvolvimento comuniário e aberto
  - Padrão Cpython
  - Multiparadigma

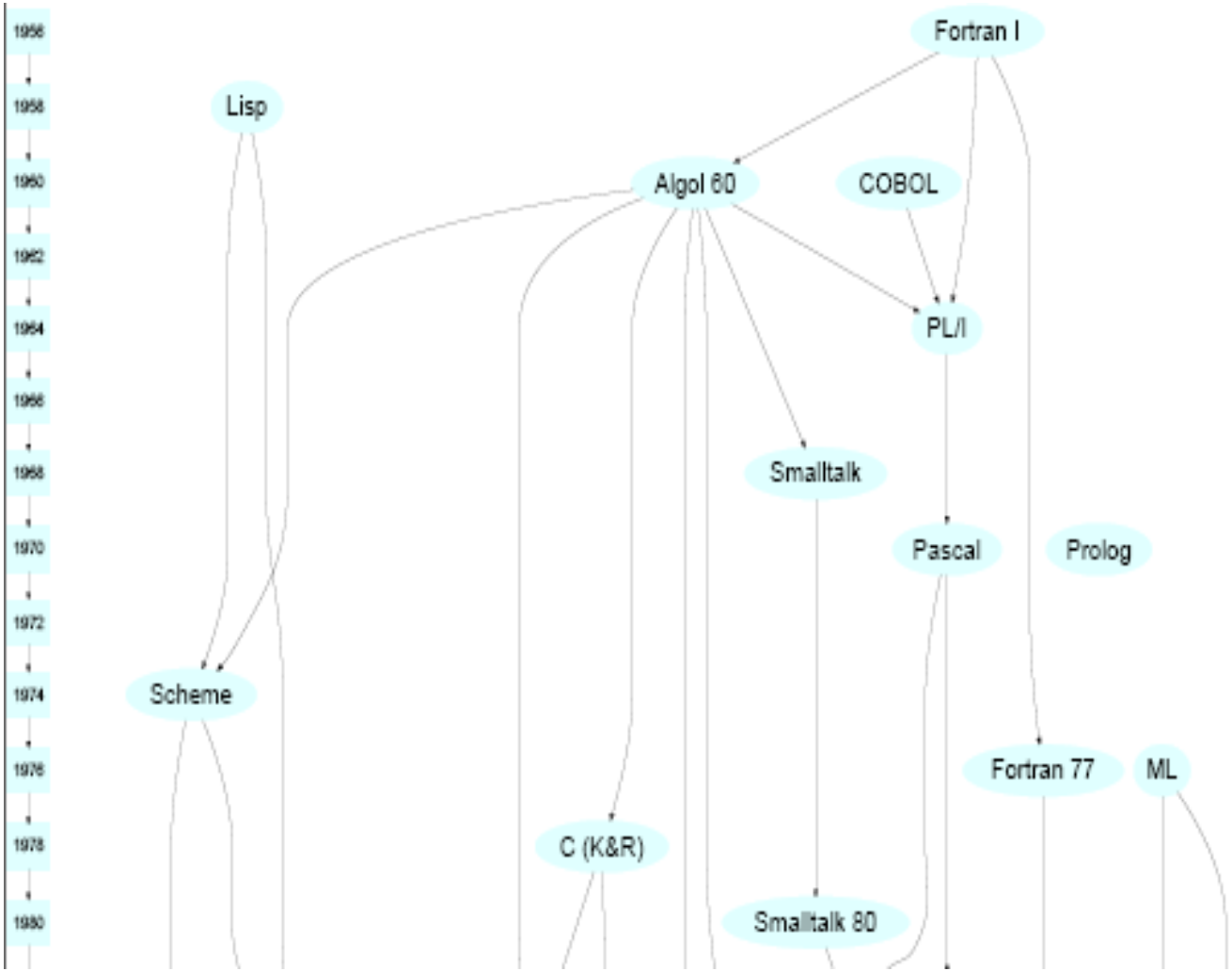
- **R**
  - 1993 – **R**oss Ihaka e **R**obert Gentleman
  - Objetivo: manipulação, análise e visualização de dados
  - Aplicação: análise estatísticas e análises de dados
  - Multiparadigma
  - Fortemente extensível (pacotes)

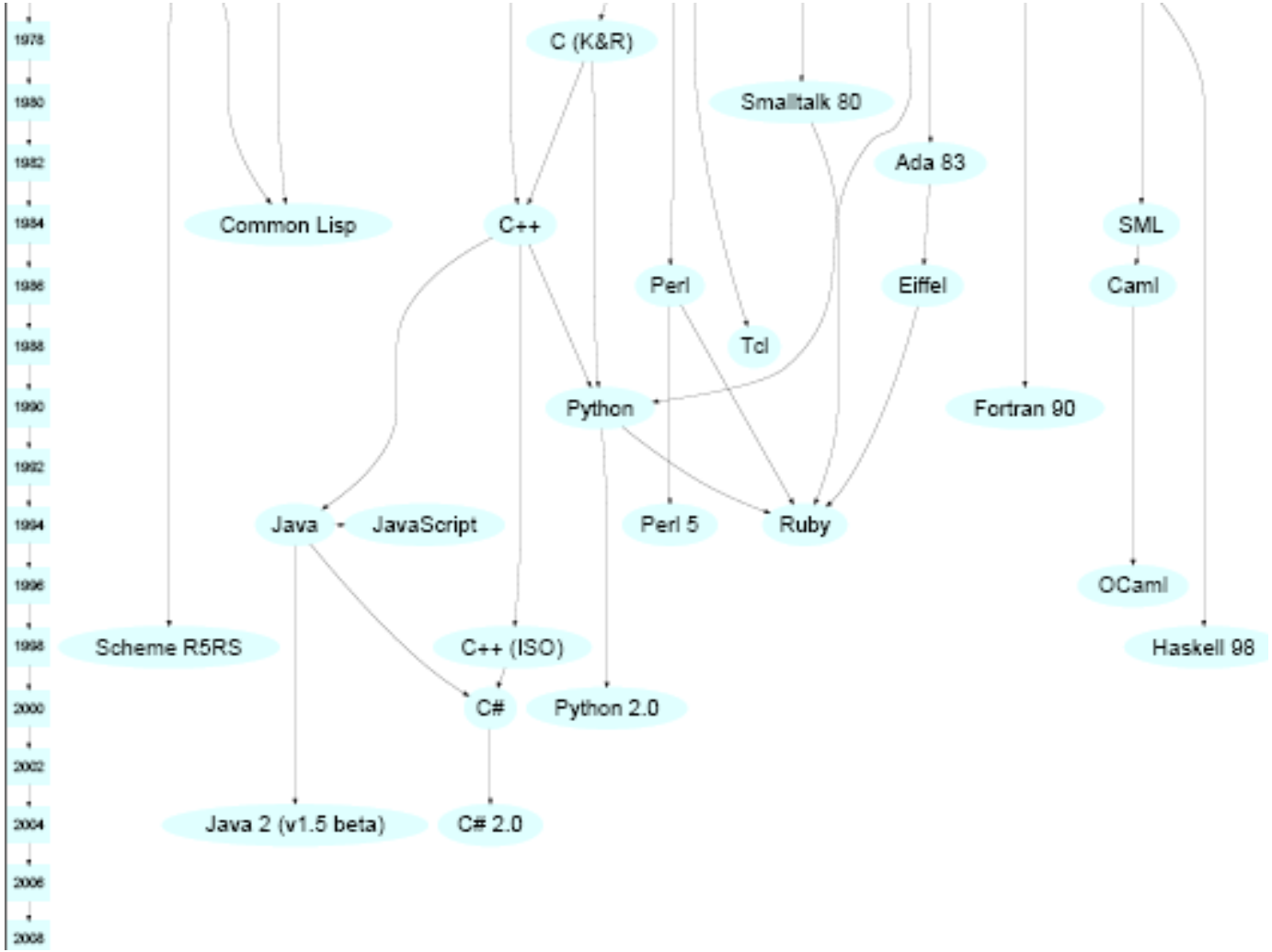
- **JAVA**
  - 1995 – Sun Microsystems Inc.
  - Objetivo: programação de dispositivos eletrônicos
  - Mais simples e mais confiável do que C++
  - Tipos (primitivos) e classes
  - Não tem ponteiros
  - Suporta herança simples
  - Criação de processos concorrentes
  - *Garbage collector*

- **JAVA**

  Avaliação:

  - Grande aceitação
  - Segura (com checagem de tipos e sem ponteiros)
  - Concorrência
  - Portabilidade? Compilada e interpretada (JVM)
  - Popularidade? Programação *web*

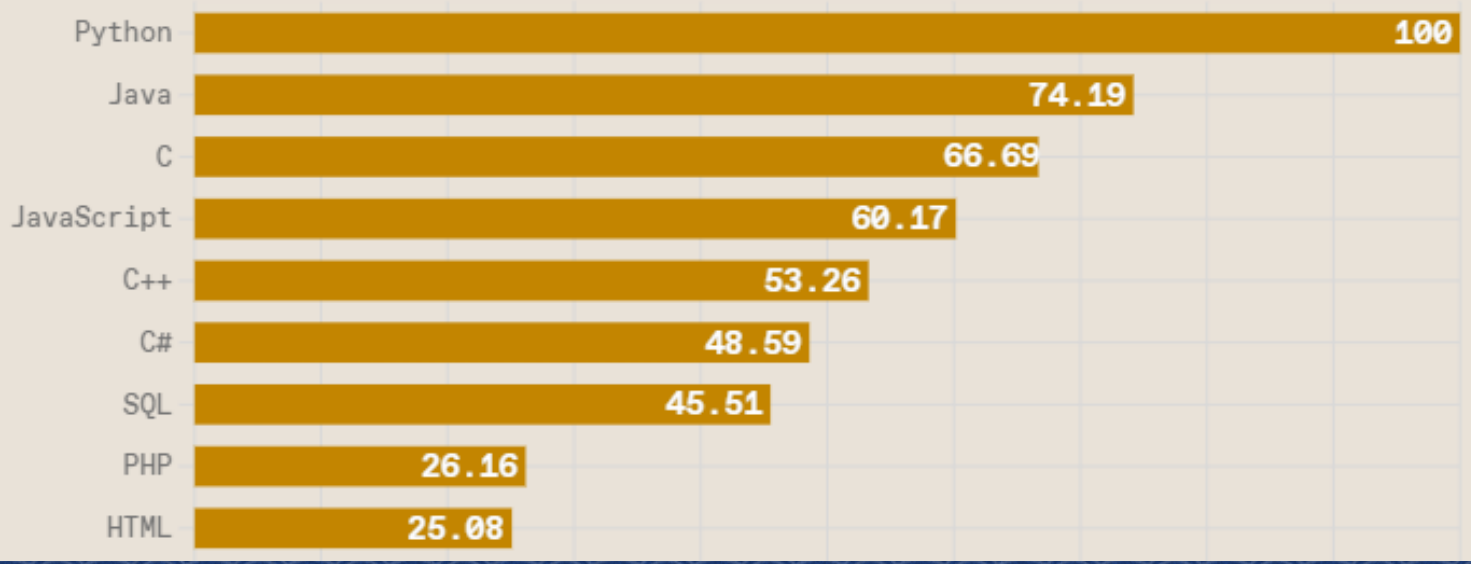

1956
1958
1960
1962
1964
1966
1968
1970
1972
1974
1976
1978
1980
Fortran I
Lisp
Algol 60
COBOL
PL/I
Smalltalk
Pascal
Prolog
Scheme
Fortran 77
ML
C (K&R)
Smalltalk 80


1978
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
2004
2006
2008
C (K&R)
Smalltalk 80
Ada 83
Common Lisp
C++
SML
Perl
Eiffel
Caml
Tcl
Python
Fortran 90
Java
JavaScript
Perl 5
Ruby
OCaml
Scheme R5RS
C++ (ISO)
Haskell 98
C#
Python 2.0
Java 2 (v1.5 beta)
C# 2.0

Top Programming Languages 2022

Click a button to see a differently weighted ranking

Spectrum | Jobs | **Trending**

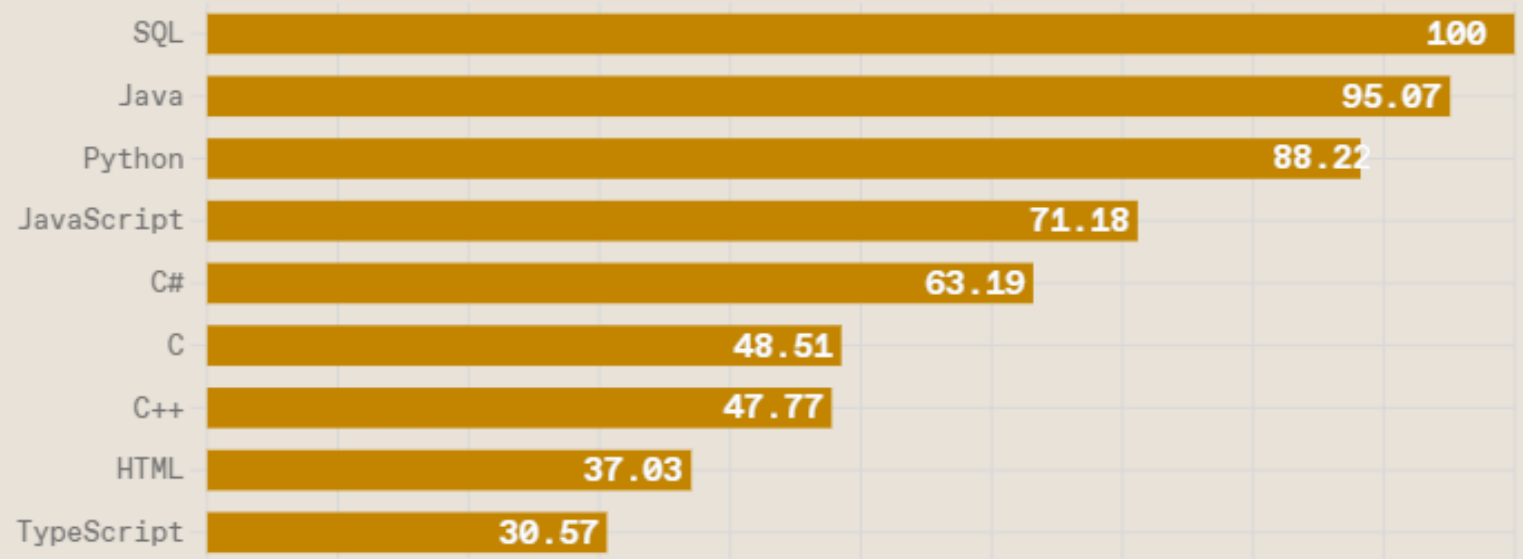


Fonte: IEEE

Top Programming Languages 2022

Click a button to see a differently weighted ranking

Spectrum | **Jobs** | Trending



Fonte: IEEE

Linguagens históricas

ALGOL - Assembly - B - BASIC - BCPL - Clipper - COBOL - Modula - Simula - Fortran

Linguagens acadêmicas

Pascal - Prolog - Lisp - Logo - Haskell - Ocaml

Linguagens comerciais

ABAP - Ada - AWK - C - C++ - C# - COBOL - Coldfusion - Delphi - Eiffel - Fortran - Lisp - Smalltalk - SQL - Visual Basic - PowerBuilder

Linguagens livres

PHP - Python - Tcl - Perl - Java - JavaScript - Shell script - Lua - Ruby

Linguagens esotéricas

BIRL – Égua - BF